



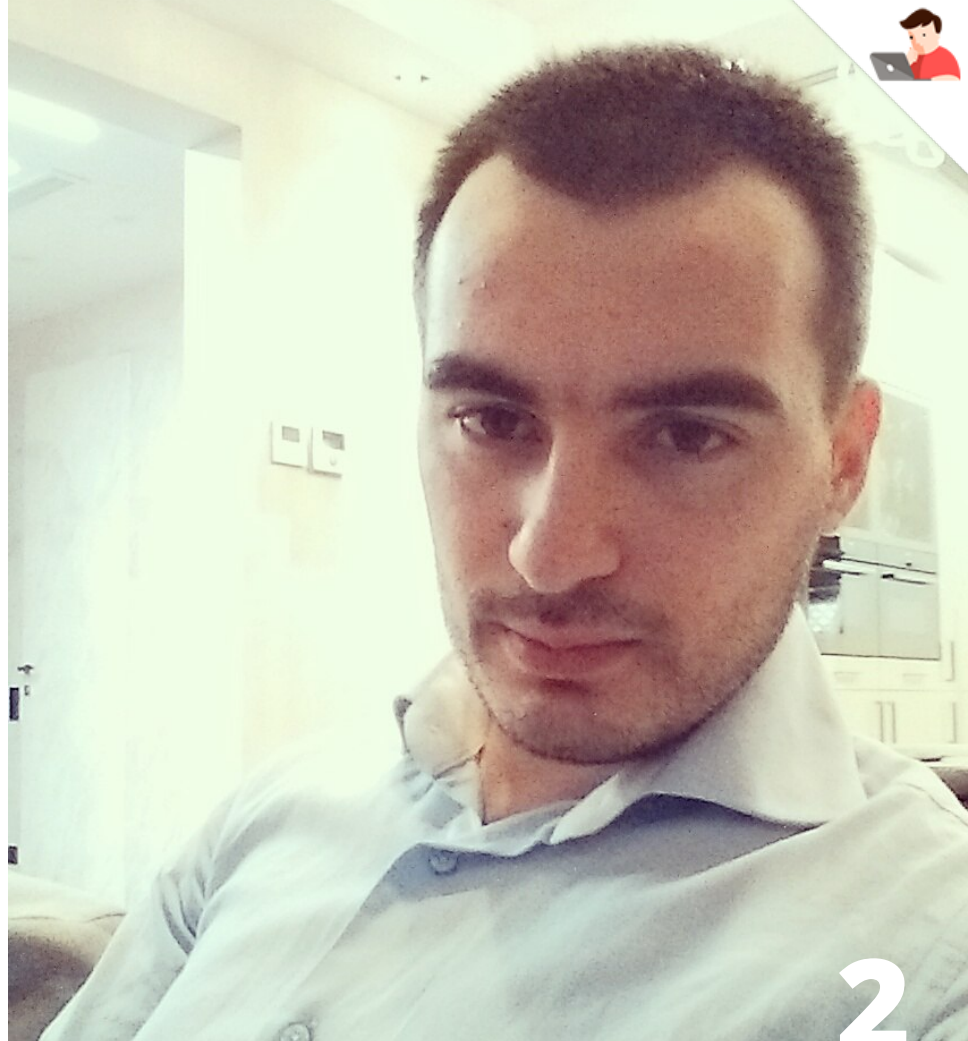
Document Object Model

Ведущий вебинара

Сергей Мелюков

Круг интересов: Backend, Frontend, GameDev, MobileDev

Место работы: Frontend разработчик профессиональных инструментов Avito



Содержание



1. Что такое DOM
2. Из чего состоит DOM
3. Методы узлов
4. Свойства узлов
5. Работа с элементами формы
6. jQuery

The background features a stylized illustration of mountains in shades of gray and white, with a white cloud on the left side. There are also yellow L-shaped corner brackets in the top right and bottom left corners.

Что такое DOM

Что такое DOM

JavaScript это *язык программирования, который не знает о том, где и как его будут использовать.*

DOM - это всего лишь *набор объектов* и методов в них, для работы с содержимым *html-страниц*.

Это своеобразная **прослойка** между JS-кодом и содержимым html-страницы





Что является содержимым html-страницы?

HTML-документ представляет собой набор узлов (**Nodes**)

Различают **3** основных типа узлов:

- элемент
- текст
- комментарии

Для составления документа, узлы вкладывают друг в друга, образуя дерево из узлов.



Из чего состоит элемент?

- открывающий тег(и атрибуты)
- дочерние элементы
- закрывающий тег

```
<div style="..." data-test="...">  
  <p>текст</p>  
</div>
```

У каждого элемента, в открывающем теге, могут быть *атрибуты*.

Атрибуты - это произвольный набор свойств(имя=значение).

Часть атрибутов являются *служебными* и влияют на поведение/отображение элемента в браузере. Остальные можно использовать в своих целях.

У некоторых элементов может быть только *открывающий* тег

Вложенность узлов

Любые узлы могут быть вложены только в **элементы**.

Другими словами, можно вложить узел типа “**текст**” в узел типа “**элемент**”, но не наоборот!

Вложенные узлы, по отношению к **родительским**, еще называют **дочерними**.

*Каждый узел знает и о своем **узле-родителе**, а узлы типа “**элемент**”, знают еще и о своих **дочерних** узлах.*

*Каждый узел знает о своих **соседних** узлах, расположенных справа и слева.*





Основные служебные атрибуты

id	уникальный идентификатор элемента (элементы, с <i>одинаковым id</i> , создавать <i>не рекомендуется</i>)
name	имя элементам (элементы, с <i>одинаковым name</i> , создавать <i>не рекомендуется</i> , за одним исключением, которое здесь не рассматривается)
class	произвольная метка элемента. У каждого элемента может быть <i>несколько</i> таких меток, разделенных <i>пробелом</i> . Разрешено использовать <i>одинаковые</i> метки для разных элементов. В основном используется для объединения элементов в <i>группы</i> . Например, для изменения визуальных стилей у всей группы элементов



Работа с DOM в JS - поиск элементов

Все функции для поиска *элементов*, хранятся внутри объекта **document**:

getElementById(id)

найти элемент по его **id-атрибуту**

getElementsByName(name)

найти элемент по его **name-атрибуту**

getElementsByClassName(className)

найти элемент по его **классу**

getElementsByTagName(tagName)

найти элемент по имени его **тега**

querySelector(selector)

найти **первый** элемент, подходящий под **css-селектор**

querySelectorAll(selector)

найти **все** элементы, подходящие под **css-селектор**

The background features a stylized illustration of mountains in shades of gray and white, with a white cloud on the left side. There are also yellow L-shaped decorative elements in the top right and bottom left corners.

Свойства и методы Node



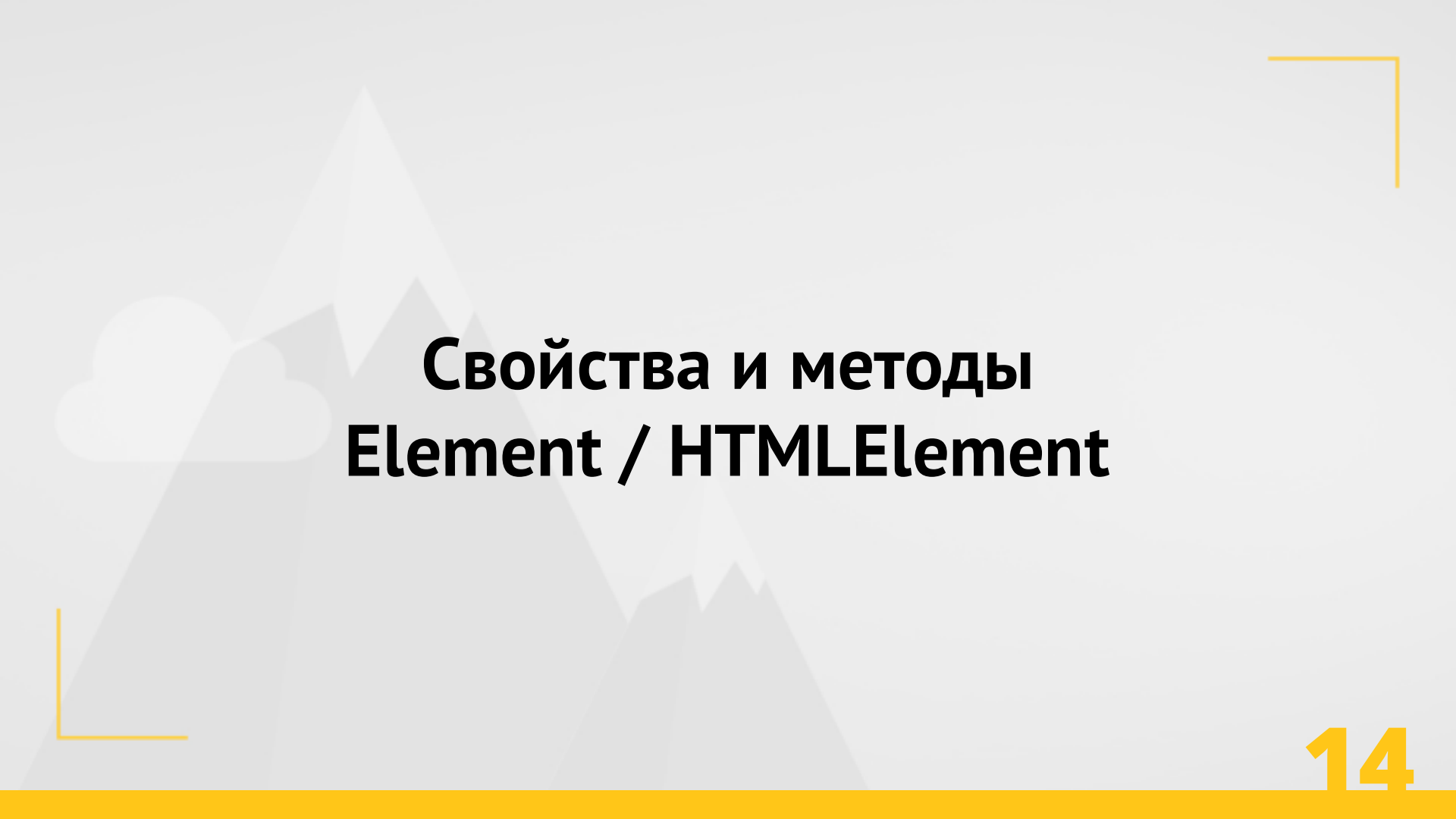
Свойства и методы Node

<code>childNodes</code>	все прямые потомки узла
<code>firstChild</code>	первый прямой потомок узла
<code>lastChild</code>	последний прямой потомок узла
<code>nextSibling</code>	соседний узел, стоящий перед текущим
<code>previousSibling</code>	соседний узел, стоящий за текущим
<code>nodeType</code>	тип узла (текст, элемент, комментарий и т.д.)
<code>parentNode</code>	родительский узел



Свойства и методы Node

<code>nodeValue</code>	для узлов типа “ текст ” и “ комментарии ”, хранит их содержимое
<code>textContent</code>	все дочерние текстовые узлы, объединенные вместе
<code>appendChild(child)</code>	добавить потомка у узлу (в конец узла)
<code>insertBefore(what, before)</code>	добавить потомка в узел, перед другим потомком внутри узла
<code>removeChild(child)</code>	удалить дочерний узел

The background features a stylized illustration of mountains in shades of gray and white, with a single white cloud on the left side. There are also yellow L-shaped decorative elements in the top right and bottom left corners.

Свойства и методы Element / HTMLElement



Свойства и методы Element / HTMLElement

id	содержит значение атрибута id
children	все прямые потомки узла, имеющие тип “ элемент ”
className	содержит значение атрибута class
firstElementChild	первый прямой потомок узла, имеющий тип “ элемент ”
lastElementChild	последний прямой потомок узла, имеющий тип “ элемент ”
nextElementSibling	соседний “ элемент ”, стоящий перед текущим
previousElementSibling	соседний “ элемент ”, стоящий за текущим



Свойства и методы Element / HTMLElement

innerHTML	html-код содержимого текущего элемента
outerHTML	html-код и элемента и его содержимого
Style	позволяет добавлять/удалять/просматривать CSS-стили элемента
tagName	имя тега узла
getAttribute(name)	получить значение атрибута по его имени
setAttribute(name, value)	установить значение атрибута по его имени
removeAttribute(name)	удалить атрибут по его имени

Свойства и методы Element / HTMLElement



remove()

удалить элемент(вместе со всем его содержимым) из его родителя

querySelector(selector)

найти первый элемент, подходящий под css-селектор

querySelectorAll(selector)

найти все элементы, подходящие под css-селектор



Работа с формами



Свойства и методы Element / HTMLElement

`document.forms` список форм на странице

`document.forms.elements` список элементов на форме

Каждый элемент формы доступен по его атрибуту **name**

`input.value` значение текстового поля

`checkbox.checked` булевый флаг

`radio.value` значение набора radio-кнопки

`select.value` значение **select box**

IFrame

iframe - это специальный узел, позволяющий вставлять один html-документ в другой.

Для корректной работы требуется указать url в атрибуте src.

Доступ к объекту window iframe, можно получить через свойство **contentWindow** данного iframe.





jQuery

jQuery



jQuery - кроссбраузерный js-фреймворк, оказавший большое влияние на снижение порога вхождения в JS для новых разработчиков.

Некоторые примеры использования:

\$(селектор)	выбрать элементы, соответствующие селектору
---------------------	---

\$(селектор).attr(атрибут, значение)

\$(селектор).addClass(класс)	добавить класс
-------------------------------------	----------------

\$(селектор).animate(параметры, скорость)	анимировать элементы
--	----------------------

Время ваших вопросов